Data Definition Modules - DDMs

The following topics are covered:

- Use of Data Definition Modules
- Listing/Displaying DDMs
- Components of a DDM

Use of Data Definition Modules

For Natural to be able to access a database file, a logical definition of the physical database file is required. Such a logical file definition is called a data definition module (DDM).

The DDM contains information about the individual fields of the file - information which is relevant for the use of these fields in a Natural program. A DDM constitutes a logical view of a physical database file.

For each physical file of a database, one or more DDMs can be defined.
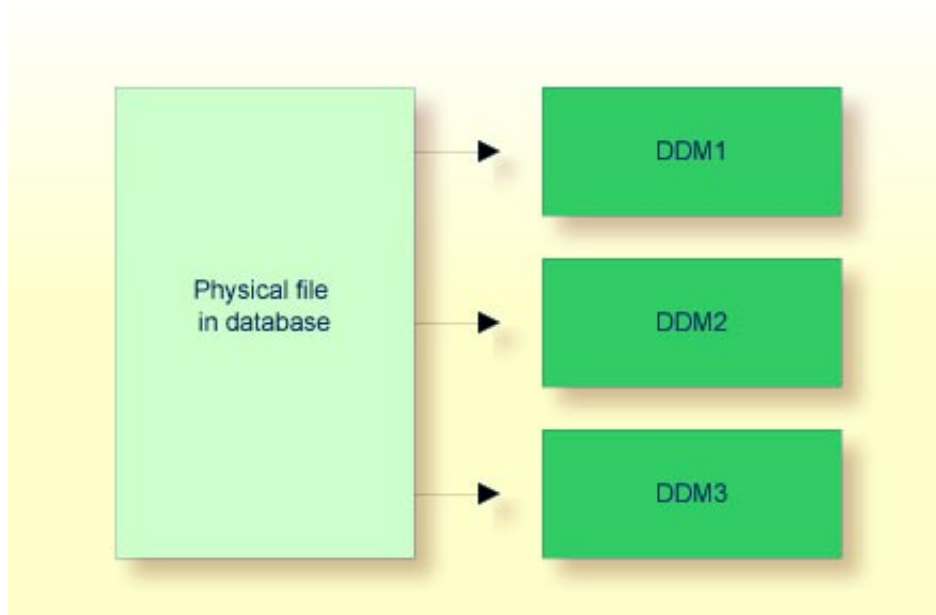
DDMs are defined by the Natural administrator with Predict (or, if Predict is not available, with the corresponding Natural function).

Use the system command SYSDDM to invoke the SYSDDM utility. The SYSDDM utility is used to perform all functions needed for the creation and maintenance of Natural data definition modules.

For further information on the SYSDDM utility, see the section SYSDDM Utility in the Natural Utilities documentation and the section DDM Services in the Natural Editors documentation.

The length of a DDM name is restricted to 32 characters.

For each database field, a DDM contains the database-internal field name as well as the "external" field name, that is, the name of the field as used in a Natural program. Moreover, the formats and lengths of the fields are defined in the DDM, as well as various specifications that are used when the fields are output with a DISPLAY or WRITE statement (column headings, edit masks, etc.).

Listing/Displaying DDMs

If you do not know the name of the DDM you want, you can use the system command LIST DDM to get a list of all existing DDMs that are available. From the list, you can then select a DDM for display.

To display a DDM whose name you know, you use the system command LIST DDM *ddm-name*.

For example:

```
LIST DDM EMPLOYEES
```

A list of all fields defined in the DDM will then be displayed, along with information about each field, see the following section Components of a DDM.

Components of a DDM

For each field, a DDM contains the following information:

Column	Explanation
T	The *type* of the field: *blank* Elementary field. This type of field can have only one value within a record. **M** Multiple-value field. This type of field can have more than one value within a record. **P** Periodic group. A periodic group is a group of fields that can have more than one occurrence within a record. **G** Group. A group is a number of fields defined under one common group name. This makes it possible to reference several fields collectively by using the group name instead of the names of all the individual fields. ***** Comment line.
L	The *level* number assigned to the field. Levels are used to indicate the structure and grouping of the field definitions. This is relevant with view definitions, redefinitions and field groups.
DB	The two-character database-*internal field name*.
Name	The 3- to 32-character *external field name*. This is the field name used in a Natural program to reference the field. **HD=** indicates a default column header to appear above the field when the field is output via a DISPLAY statement. If no header is specified, the field name is used as column header. **EM=** indicates a default edit mask to be used when the field is output via a DISPLAY statement.
F	The *format* of the field (A=alphanumeric, N=numeric unpacked, P=packed numeric, etc.).
Len	The *length* of the field. For numeric fields, length is specified as "*nn.m*", where "*nn*" is the number of digits before the decimal point and "*m*" is the number of digits after the decimal point.

Column	Explanation
S	The type of *suppression* assigned to the field: **N** indicates *null-value suppression*, which means that null values for the field will not be returned when the field is used to construct a basic search criterion (WITH clause of a FIND statement), in a HISTOGRAM statement, or in a READ LOGICAL statement. **F** indicates that the field is defined with the *fixed storage* option (that is, the field is not compressed). A blank indicates *normal compression*, which means that trailing blanks in alphanumeric fields and leading zeros in numeric fields are suppressed.
D	The *descriptor* type of the field; for example: **D** elementary descriptor, **N** non-descriptor, **P** phonetic descriptor. **U** subdescriptor, **S** superdescriptor, A blank in this column indicates that the field is not a descriptor. A descriptor can be used as the basis of a database search. A field which has a "D" or "S" in this column can be used in the BY clause of the READ statement. Once a record has been read from the database using the READ statement, a DISPLAY statement can reference any field which has either a "D" or a blank in the "D" column.
Remarks	This column can contain *comments* about the field.

Above the list of fields, the following is displayed: the number of the file from which the DDM is derived (DDM FNR), the number of the database where that file is stored (DDM DBID), and the "Default Sequence" field, that is, the name of the field used to control logical sequential reading of the file if no such field is specified in the READ LOGICAL statement of a program.